

CLAIMS

What is claimed is:

- 1 1. A method for reducing the recovery time after a failure, comprising the steps of:
2 maintaining a checkpoint value that indicates which records of a plurality of records
3 have to be processed after the failure, wherein the plurality of records indicate
4 changes for a plurality of data blocks;
5 determining a target checkpoint value based on a desired number of data block reads
6 that will be required during a redo phase of recovery; and
7 updating the checkpoint value based on the target checkpoint value.
- 1 2. The method of Claim 1, further comprising the step of:
2 maintaining, in volatile memory, one or more sorted buffer queues, wherein each
3 sorted buffer queue includes queue entries that are inserted into said sorted
4 buffer queue based on an index value associated with said queue entry,
5 wherein each queue entry reflects a change to a data block of the plurality of data
6 blocks.
- 1 3. The method of Claim 2, wherein the one or more sorted buffer queues are one or
2 more circular sorted buffer queues, and wherein a modulus operation is used identity
3 the index value associated with each circular sorted buffer queue when inserting a
4 queue entry into the circular sorted buffer queue.
- 1 4. The method of Claim 2, further comprising:
2 maintaining a count of the queue entries in each of the one or more sorted buffer
3 queues.

- 1 5. The method of Claim 2, wherein each of the one or more sorted buffer queues is
2 associated with a target number of queue entries, and the method further comprises:
3 determining a count of the queue entries in each of the one or more sorted buffer
4 queues; and
5 if the count of the queue entries in a particular sorted buffer queue of the one or more
6 sorted buffer queues is greater than target number of queue entries associated
7 with the particular sorted buffer queue, then reducing the number of queue
8 entries in the particular sorted buffer queue to the target number of queue
9 entries associated with the particular sorted buffer queue.
- 1 6. The method of Claim 2, wherein the step of updating the checkpoint value comprises:
2 updating the checkpoint value to equal a byte offset in a redo log associated with the
3 queue entry in the one or more sorted buffer queues that is associated with the
4 least recently modified buffer in any queue entry in the one or more sorted
5 buffer queues.
- 1 7. The method of Claim 1, further comprising the step of:
2 maintaining, in volatile memory, one or more partially sorted buffer queues, wherein
3 each partially sorted buffer queue includes queue entries that are inserted into
4 said partially sorted buffer queue based on an index value associated with said
5 queue entry,
6 wherein each queue entry reflects a change to a data block of the plurality of data
7 blocks.
- 1 8. The method of Claim 1, wherein the checkpoint value is a first checkpoint value in a
2 plurality of checkpoint values, and the method further comprises the step of:
3 maintaining the plurality of checkpoint values, wherein each of the plurality of
4 checkpoint values indicates which records of the plurality of records have to

5 be processed after the failure by a particular database server accessing a
6 database, wherein the plurality of records indicate changes for the plurality of
7 data blocks.

1 9. A method for controlling an amount of time that is needed to recover after the
2 occurrence of a database system failure, the method comprising the steps of:
3 maintaining a checkpoint value that indicates which records of a plurality of records
4 have to be processed after the failure, wherein the plurality of records indicate
5 changes for a plurality of data blocks;
6 determining a required recovery time, wherein the required recovery time indicates a
7 maximum length of time that is to be allowed for recovering after said
8 database system failure;
9 determining a maximum number of data block reads that can be performed within the
10 required recovery time; and
11 periodically advancing the target checkpoint value based on the maximum number of
12 data block reads that can be performed within the required recovery time.

1 10. A computer-readable medium carrying one or more sequences of instructions for
2 reducing the recovery time after a failure, wherein execution of the one or more
3 sequences of instructions by one or more processors causes the one or more
4 processors to perform the steps of:
5 maintaining a checkpoint value that indicates which records of a plurality of records
6 have to be processed after the failure, wherein the plurality of records indicate
7 changes for a plurality of data blocks;
8 determining a target checkpoint value based on a desired number of data block reads
9 that will be required during a redo phase of recovery; and
10 updating the checkpoint value based on the target checkpoint value.

1 11. The computer-readable medium of Claim 10, wherein execution of the one or more
2 sequences of instructions by the one or more processors causes the one or more
3 processors to further perform the step of:

4 maintaining, in volatile memory, one or more sorted buffer queues, wherein each
5 sorted buffer queue includes queue entries that are inserted into said sorted
6 buffer queue based on an index value associated with said queue entry,
7 wherein each queue entry reflects a change to a data block of the plurality of data
8 blocks.

1 12. The computer-readable medium of Claim 11, wherein the one or more sorted buffer
2 queues are one or more circular sorted buffer queues, and wherein a modulus
3 operation is used identity the index value associated with each circular sorted buffer
4 queue when inserting a queue entry into the circular sorted buffer queue.

1 13. The computer-readable medium of Claim 11, wherein execution of the one or more
2 sequences of instructions by the one or more processors causes the one or more
3 processors to further perform the steps of:
4 maintaining a count of the queue entries in each of the one or more sorted buffer
5 queues.

1 14. The computer-readable medium of Claim 11, wherein each of the one or more sorted
2 buffer queues is associated with a target number of queue entries, and wherein
3 execution of the one or more sequences of instructions by the one or more processors
4 causes the one or more processors to further perform the steps of:
5 determining a count of the queue entries in each of the one or more sorted buffer
6 queues; and
7 if the count of the queue entries in a particular sorted buffer queue of the one or more
8 sorted buffer queues is greater than target number of queue entries associated

9 with the particular sorted buffer queue, then reducing the number of queue
10 entries in the particular sorted buffer queue to the target number of queue
11 entries associated with the particular sorted buffer queue.

1 15. The computer-readable medium of Claim 11, wherein the step of updating the
2 checkpoint value comprises:
3 updating the checkpoint value to equal a byte offset in a redo log associated with the
4 queue entry in the one or more sorted buffer queues that is associated with the
5 least recently modified buffer in any queue entry in the one or more sorted
6 buffer queues.

1 16. The computer-readable medium of Claim 10, wherein execution of the one or more
2 sequences of instructions by the one or more processors causes the one or more
3 processors to further perform the step of:
4 maintaining, in volatile memory, one or more partially sorted buffer queues, wherein
5 each partially sorted buffer queue includes queue entries that are inserted into
6 said partially sorted buffer queue based on an index value associated with said
7 queue entry,
8 wherein each queue entry reflects a change to a data block of the plurality of data
9 blocks.

1 17. The computer-readable medium of Claim 10, wherein the checkpoint value is a first
2 checkpoint value in a plurality of checkpoint values, and wherein execution of the one
3 or more sequences of instructions by the one or more processors causes the one or
4 more processors to further perform the step of:
5 maintaining the plurality of checkpoint values, wherein each of the plurality of
6 checkpoint values indicates which records of the plurality of records have to
7 be processed after the failure by a particular database server accessing a

8 database, wherein the plurality of records indicate changes for the plurality of
9 data blocks.

1 18. A computer-readable medium carrying one or more sequences of instructions for
2 controlling an amount of time that is needed to recover after the occurrence of a
3 database system failure, wherein execution of the one or more sequences of
4 instructions by one or more processors causes the one or more processors to perform
5 the steps of:
6 maintaining a checkpoint value that indicates which records of a plurality of records
7 have to be processed after the failure, wherein the plurality of records indicate
8 changes for a plurality of data blocks;
9 determining a required recovery time, wherein the required recovery time indicates a
10 maximum length of time that is to be allowed for recovering after said
11 database system failure;
12 determining a maximum number of data block reads that can be performed within the
13 required recovery time; and
14 periodically advancing the target checkpoint value based on the maximum number of
15 data block reads that can be performed within the required recovery time.